

**This Page Is Inserted by IFW Operations
and is not a part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- **BLACK BORDERS**
- **TEXT CUT OFF AT TOP, BOTTOM OR SIDES**
- **FADED TEXT**
- **ILLEGIBLE TEXT**
- **SKEWED/SLANTED IMAGES**
- **COLORED PHOTOS**
- **BLACK OR VERY BLACK AND WHITE DARK PHOTOS**
- **GRAY SCALE DOCUMENTS**

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



EP0875843

Biblio

Desc

Claims

Page 1

Drawing


esp@cenet



Method and apparatus for inserting advertisements and the like in a data stream of an internetworked client for display during the void time

Patent Number: ☐ EP0875843
Publication date: 1998-11-04
Inventor(s): GILMORE BADEN EDWARD (US); REID CHRISTOPHER (US)
Applicant(s): ADLETTS INTERNATIONAL LLC (US)
Requested Patent: ☐ JP10320336
Application Number: EP19980302857 19980414
Priority Number(s): US19970837157 19970414
IPC Classification: G06F17/30
EC Classification: G06F17/60B2
Equivalents: AU6078598
Cited Documents: US5572643; WO9707656; WO9630864; JP9091215

Abstract

An improved hypertext viewing system provides inserts to a user at a browser client to view while a requested page loads. The inserts can be any material, but one type of material being animated advertisements. With animated ads, the ad can be downloaded in much less time than it takes to show the ad. This is preferred, because it frees the network to be used to download a requested page between the time that the ad finished downloading and finishes playing. Of course, where the network is much faster than the source of the requested page, such a delay is less of an issue. 

Data supplied from the esp@cenet database - 12

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-320336

(43) 公開日 平成10年(1998)12月4日

(51) Int.Cl. ⁶	識別記号	F I
G 0 6 F 13/00	3 5 5	G 0 6 F 13/00 3 5 5
15/00	3 1 0	15/00 3 1 0 A
17/30		15/40 3 1 0 F
		15/403 3 4 0 A

審査請求 未請求 請求項の数9 O L (全 11 頁)

(21) 出願番号 特願平10-102329

(22) 出願日 平成10年(1998)4月14日

(31) 優先権主張番号 08/837157

(32) 優先日 1997年4月14日

(33) 優先権主張国 米国 (U S)

(71) 出願人 598049492

アドレッツ インターナショナル リミテ
ッド ライアビリティ カンパニー
アメリカ合衆国 カリフォルニア州
94965 ソーサリット ブリッジウェイ
3020 スイート 600

(72) 発明者 バーデン エドワード マックマスター
ギルモア

アメリカ合衆国 カリフォルニア州
94965 ソーサリット ジョンソン スト
リート 410エイ

(74) 代理人 弁理士 中村 稔 (外6名)

最終頁に続く

(54) 【発明の名称】 空き時間中に表示する目的でインターネットにネットワーク接続されたクライアントのデータストリーム内に広告等を挿入するための方法及び装置

(57) 【要約】

【課題】 利用可能なネットワーク帯域幅を効率良く使用し、この空き時間の存在により提供される機会を効果的に使用する。

【解決手段】 改良型ハイパーテキストビューイングシステムが、リクエストされたページをロードする間に見るようにブラウザクライアントの前にいるユーザに対してインサートを提供する。このインサートはいかなる材料であってもよいが、1つの材料タイプとしては、アニメ式広告がある。アニメ式広告の場合、広告は、それを示すのにかかる時間よりもはるかに短時間でダウンロードされ得る。こうして広告がダウンロードし終えた時点と放映し終る時点の間にリクエスト対象のページをダウンロードするのに使用するべくネットワークが解放されることから、これは好ましいことである。当然のことながら、ネットワークがリクエスト対象のページのソースに比べはるかに高速である場合、このような遅延はさほど問題とはならない。

【特許請求の範囲】

【請求項1】 目標ページのリクエストと目標ページの提示の間で非ゼロ時間が経過する、ハイパーテキストウェブを拾い読みする方法において、ブラウザクライアントから中間サーバまでページリクエストを送る段階；中間サーバによるリクエストの受理後、ブラウザクライアントに対し中間プレゼンテーションを表わすデータブロックを送る段階；ブラウザクライアントに結合された表示位置上で中間プレゼンテーションを提示する段階；及び中間プレゼンテーションの完了に先立ち、表示装置上の表示のため、ブラウザクライアントに対し目標サーバから目標ページをリクエストする段階；を含んで成る方法。

【請求項2】 中間プレゼンテーションを提示する段階が、広告を提示する段階である請求項1に記載の方法。

【請求項3】 広告は、複数の広告のうちから選択された1つであり、選択される広告はブラウザのタイプ、クライアントネットワークアドレス又はユーザーに関する人口統計学的情報に応じたものである、請求項2に記載の方法。

【請求項4】 プレゼンテーションがアニメーションである、請求項1に記載の方法。

【請求項5】 中間サーバからクライアントブラウザまで目標ページに対する参照を送る段階；及びクライアントブラウザから目標ページのサービスを提供をする目標サーバまでリファレンスを送る段階；が、目標ページリクエスト段階に含まれている請求項1に記載の方法。

【請求項6】 目標ページのリクエストと目標ページの提示の間で非ゼロ時間が経過する、ハイパーテキストウェブを拾い読みする方法において、ブラウザクライアントから中間サーバまでページリクエストを送る段階；中間サーバによるリクエスト受理後、ブラウザクライアントに対し中間プレゼンテーションを表わすデータブロックを送る段階；ブラウザクライアントに結合された表示装置上で中間プレゼンテーションを提示する段階；データブロックの送信と実質的に同時に、目標サーバに対し目標ページのリクエストを送る段階；及び表示装置上での表示のため、ブラウザクライアントに対して目標サーバからリクエストされたページを送る段階；を含んで成る方法。

【請求項7】 目標ページのリクエストを送る段階が、中間サーバからリクエストを送る段階である、請求項6に記載の方法。

【請求項8】 目標ページのリクエストを送る段階が、クライアントブラウザからリクエストを送る段階である、請求項6に記載の方法。

【請求項9】 クライアントブラウザからリクエストを送る段階の前には中間サーバからクライアントブラウザまでリクエストを送る段階が先行している、請求項8に

記載の方法。

【発明の詳細な説明】**【0001】**

【産業上の利用分野】 本発明は、ネットワーク化された環境内でクライアントシステムがサーバシステムからの応答を待っている間に、この環境内でクライアントシステム上に広告といったような情報を獲得し表示するためのシステムに関する。

【0002】

【従来の技術】 近年、インターネットといったような大規模ネットワークシステムの使用が劇的に増大してきた。これを利用すると、多くのビューワ（観察者）がネットワーク上のサイト及びページを見ることが可能となる一方で、利用の増大によって、或るページに関するクライアントのリクエストに対するサーバ応答の遅延が大きくなってきている。このため、ユーザは往々にして拾い読み（ブラウジング）における多大な遅延を経験することになり、ここでユーザはサーバ応答を待つこと以外何もすることのない状態に放置される。「拾い読み（ブラウジング）」とは、大域ワールドワイドウェブ（「WWW」）といったようなハイパーテキスト文書ウェブとユーザーとの標準的対話を説明するものである。WWWを拾い読みするためには、ユーザはブラウザを実行し、このブラウザがユーザに最初のページを提示する。この最初のページは、何らかのユーザー命令が無い場合にブラウザが表示する省略時解釈ページであるか、或いは又ユーザにより指定される汎用資源ロケータ（「URL」）によって指定される特定のディレクトリ内の特定のサーバ上の1ページである。ブラウザの中に表示される標準的ハイパーテキストページは、その他のハイパーテキストページへの一定数の「リンク」を含んでいる。標準的ブラウザは、ブラウザメモリ内にリンクされたページのURLを維持する一方で、下線がどこされ全く異なるカラーのテキストとしてリンクのアンカー部分を示す。ユーザが通常はマウスカーソルをアンカーテキスト上でクリックすることによって1つのリンクを選択した時点で、ブラウザは、選択されたリンクのURLにより指定されたページについてのリクエストを送り出すことになる。さまざまな理由から、次に選択されたページが直ちに現われるわけではない。この遅延は、インターネットへのユーザの接続に対する帯域幅の制限条件、インターネットの混雑又は、リンクすべきページを提供するはずのサーバからの低い応答時間に起因すると考えられる。この遅延は、きわめて一般的なものであるため、大部分のブラウザは、ブラウザがサーバからのページを待機していることをユーザに表示するためのアニメーションアイコンを内含しており、かくして、ブラウザが単に作動を停止したという印象をユーザがもたないようにしている。ウェブページをリクエストする時点と受理する時点の間のこのアイドル時間つまり「空き時間」は、

1分以上といった長いものになることがある。これまでのところ、この空き時間は、大部分のユーザが無為に応答を待つ時間であった。その上、遅延がサーバにおけるものである場合、ユーザとインターネットの間のネットワーク帯域幅は、使われてさえいないのである。

【0003】

【発明が解決しようとする課題】従って必要とされているのは、利用可能なネットワーク帯域幅を効率良く使用し、この空き時間の存在により提供される機会を効果的に使用するためのシステムである。

【0004】

【課題を解決するための手段】本発明に従うと、改良型ハイパーテキストビューイングシステムが提供されている。1つの実施形態においては、リクエストされたページがロードされる間に見るべくブラウザクライアントにおいてユーザに対しインサートが提供される。これらのインサートは、いかなる材料であってもよいが、1つの材料タイプとしては、アニメ式広告がある。アニメ式広告の場合、広告は、それを示すのにかかる時間よりもはるかに短時間でダウンロードされ得る。こうして広告がダウンロードし終えた時点と放映し終る時点の間にリクエスト対象のページをダウンロードするのに使用するべくネットワークが解放されることから、これは好ましいことである。当然のことながら、ネットワークがリクエスト対象ページのソースに比べはるかに高速である場合、このような遅延はさほど問題とはならない。本書中の本発明の内容及び利点については、明細書の残りの部分及び添付図面を参照することによってさらに理解できるものと思われる。

【0005】

【実施例】本発明は、数多くの異なる環境において利用可能であるが、その全てが本書で記述されているわけではない。ここで記述される1つの環境は、図1に示されている。この図には、インターネットにネットワークされた拾い読みシステム10が示され、ここでは、従来の要領でブラウザクライアント12が大域インターネット14に結合されている。複数のサーバ（スポンササーバ16、独立サーバ18及び広告サーバ20）もインターネット14に接続した状態で示されている。ブラウザクライアント12は、当該技術分野において周知の通り、インターネットブラウザプログラムを実行する従来のパーソナルコンピュータ（例えば Apple Computer, Sun Microsystems 製のもの及び Microsoft Corporation により開発されたウィンドウズ® オペレーティングシステムを実行するパーソナルコンピュータ）であってよい。ここで使用する例は、Java アプレット及び Java アプリケーションを実行できる Java □ 言語互換性ブラウザを基準としているが、Microsoft Corporation が開発した Active X □ API テクノロジーといったようなその他の数多くの同等のコンフィギュレーションを使用することも

同様に可能であるということを理解すべきである。ブラウザクライアント12は、周知のトランスポート制御プロトコル及びインターネットプロトコル（「TCP/IP」）を用いて通信するべく、モデム、ISDN接続、ネットワークカードなどを用いてインターネット14に結合される。拾い読みする場合、ブラウザクライアント12は一般に、指令、リクエスト、データ及びハイパーテキストマークアップ言語（「HTML」）文書を転送するべくハイパーテキストトランスポートプロトコル（「HTTP」）を用いてインターネット14を介してサーバと通信する。図1には3タイプのサーバが示されている。一般に、これらのサーバは、リクエストに応じてクライアントに対しHTMLページのサービス提供をする（送り出す）。これらのページ（すなわち「ウェブページ」）は、サーバにおいて記憶デバイスから検索された文書であってもよいし、或いは又、ページは、特定のクライアント又はサーバの状態に結びつけられているか又はテーブルの中に記憶されている変数に基づいて動的に生成されたページであってよい。任意のサーバ上又は1つのサーバの一部分の上の1つの収集物として論理的に方向づけされた（或いはさらに多数のサーバ上に物理的に分配された）ページの収集物は往々にしてサイトと呼ばれ、又ワールドワイドウェブについて言う場合にはウェブサイトと呼ばれている。

【0006】サーバのタイプは定められておらず、時によって、及びページによって変化し得るということを理解しておくべきである。スポンササーバ16というのは、それが広告サーバ20のオペレータと共働でページのサービス提供をすることから、このように名付けられたものである。独立サーバ18は、それが本書に記述されているプロセスに参加するのに広告サーバ20を気にかける必要がないことからこう名付けられているものである。広告サーバ20は、それが空き時間中にブラウザクライアント12のユーザに対し提示される広告のサービス提供をすることからその名がついている。しかしながら、広告サーバ20が空き時間の娯楽などといったように非広告要素のサービス提供を行なうことも全く同様に可能であるということも理解すべきである。

【0007】ここで次に異なるタイプのサーバの役割について、図2に関連してウェブを拾い読みするための従来のプロセスの説明に続いて、図3を参照して説明する。図2は、クライアントブラウザ表示装置上に何が示されるか、ブラウザクライアントソフトウェアによってどんな行動がとられるか、クライアントブラウザと目標サーバの間でどのメッセージがインターネット14上に通過させられるか、そして目標サーバによりとられる行動を例示する、標準的な拾い読みプロセスのステップを示すフローチャートである。図2中の各ステップは、以下の記述の中に対応するリファレンスが示されているステップ番号（S1、S2など）でラベル付けされている。

まず最初に、クライアントブラウザは省略時解釈又は選択されたページをロードする(S1)。ブラウザに応じて、これは、開始パラメータとして指定された1ページ、コンフィギュレーションファイルの中で指定されたホームページ、ブラウザソフトウェアのライタにより指定された省略時解釈開始ページなどであってよい。しかしながら、指定された時点で、ユーザとの対話プロセスは、クライアントブラウザがブラウザ表示装置上でページを表示することから始まる(S2)。ページは、リンクがテキストの残りの部分から離してセットされた状態で表示される。かかるページの一例としては、テキストの残りの部分から離してセットするべくリンク42に下線が付された状態で図4(a)に示された「ページA」40がある。HTMLテキストは、次のようなテキストスニペットを含む：

```
Click<a href="http://site/file">here</a>;to go to Page B
```

これは次のように表示する：

Click here to go to Page B (ページBへ進むにはここをクリックせよ)。

【0008】ここにおいて、site はページBのサービスを提供するサーバで置換され、file はサーバサイト上でページBに与えられる名前である。サイト及びファイルは表示装置上に表示されないが、リファレンステーブル内にクライアントブラウザによりセーブされる。ユーザがリンク上でマウスカーソルをクリックするか(S3)、又はその他の形でリンクを指定した時点で、ブラウザクライアントはその選択を検出し、ユーザが選択したページについて目標サーバにリクエストを行なう(S4)。目標サーバはリクエストを受理し(S5)、ページリクエストを処理し始める(S6)。クライアントとインターネットの間のネットワーク接続、サーバの接続又はインターネットを通る経路が緩慢である場合、このプロセスは直ちに始まらない。サーバはページリクエストの処理が終了した時点で、インターネット上をクライアントまでリクエストされたページを送り(S7)、クライアントはそのページを受理し(S8)、それをユーザに表示する(S9)。ページの例44は、図4(b)に示されている。目標サーバが緩慢であるか又はオーバーロードされている場合、サーバの遅延はトランジット遅延に加算する。この遅延は往々にしてかなり長いものであることから、ユーザは、ブラウザソフトウェアのクライアントマシンが凍結され作動を停止したと考えるかもしれない。この印象を避けるため、ブラウザは、アニメ式アイコン、砂時計、転がるボール又はその他の待機表示を示すかもしれない。この時間中に、ユーザは一般にページBが到着するのを待つばかりである。ユーザは、空き時間が終わるまでどの位の長さがあるのかの表示を全く与えられていないため、前もって計画できない。

【0009】ここで図3を参照すると、空き時間を効率良く用いるプロセスが示されている。従来の方法の場合と同様、クライアントブラウザは、ページをロードし(S30)それを表示する(S31)ことから始めるが、表示されたページは、スポンササーバからのものである。上述のページAが表示されるべきページであるならば、表示は図4(a)中のページ40により示されているとおりに現われることになるが、ブラウザは、リンク42のために異なるリンクリファレンス(URL)を維持する。この例でURLは次の通りである：

```
<a href="http://ad site/ad.dll?id=1&goto=http://site/file">
```

なおここで ad site は、広告サーバのドメイン名であり、http://site/fileはリファレンスのためのURL(上述の例ではページB)である。URL内にはスポンサ付きサーバ識別子(「id=1」)が内含されており、そのため広告サーバは、リンクがどこから来たかを知ることになる。このことは、トラヒックを追跡するのに有用であり、スポンサ付きサーバオペレータが、それが広告サーバに送る接続の数に基づく支払いを受ける場合、スポンサ付きサーバ識別子は、収入追跡のために使用される。

【0010】ユーザが Link to ページB上でクリックする(又はその他の形でこれを選択する)と(S32)、クライアントブラウザはそのクリックを検出しそのページについてのリクエストを作成する(S33)。リンクに対するURLはページBについてのサーバではなく広告サーバをポイントするため、リクエストは広告サーバに送られる。広告サーバは、リクエストを受理した時点で(S34)広告を実行するべくコードを送り(S35)、このコードはクライアントブラウザによって処理され(S36)、広告は表示装置上で表示される(S37)。送られる指定のコードは、インプリメンテーションに応じて変わる。1つの変形形態では、広告サーバは、広告リンクの受理に応じて「クッキー」についてクライアントにたずねる。クッキーは、リクエストされるまで保持するためブラウザに与えられるデータブロック(通常は1つのテキストストリング)である。クッキーは、サーバがブラウザクライアントの援助を得てブラウザクライアントのために1つの状態を維持することができるよう用いられる。ブラウザクライアントが特定のクッキーを有しこれを提供する場合、サーバは広告コードを送るステップを飛び越し、単に目標ページへの再配置のためのリクエストを送ることになる。

【0011】使用可能なこのような2つのインプリメンテーションとしては、Java アプレット及び Java アプリケーションがある。Java アプリケーションとして広告が提示される場合、ステップS35で送られるコードはHTMLファイルとして送られる。図5は、このようなHTMLページ50の一例である。クライアントブラ

ウザは、HTMLページ50を受理した時点で、与えられたパラメータを用いて1つのウィンドウを開き、URL52によって指定されたページについての内容を検索し、それに1つの広告指定子54を渡す。ほぼ同時に、ブラウザは、ブラウザにもう1つのHTMLページ例えば図6に示されたページ60を渡すことによって目標ページまで「押され」る。ページ60は、ステップ34で受理されたもとのリクエストから得られた目標リファレンス62を内含する。その結果、広告がブラウザウィンドウ内で目標を示している一方で、ページBは目標サーバからリクエストされつつある(S38)。

【0012】同様に、ステップS36内で受理されるコードがJavaアプリケーションである場合、送られるHTMLページは図7に示されたページ70に似ている。クライアントブラウザにより受理された時点で、このクライアントブラウザは、Javaアプリケーションでイネーブルされた場合、示されたパラメータと共に新しいアプリケーションウィンドウを開くことになる。提供可能なJavaアプリケーション及びJavaアプレットの例は、付録Aに示されている。いずれの方法でも、ブラウザがひとたびコードを受理し(S36)これを表示した(S37)ならば、表示が図8に示されているページ80といったような1つのページを含んでいることを条件として、ユーザは、目標サーバが目標ページのためのリクエストを処理(S39)し目標ページ(この例ではページB)をブラウザクライアントに送っている(S40)間、広告を見る。ページBが場合によって目標サーバから受理された時点で(S41)、それは、広告ウィンドウが開かれる前に存在していたウィンドウの中で表示される(S42)。広告はそれ自体のウィンドウの中にあることから、ページBが受理され表示されている時にはまだ放映中である(S37')。いくつかのケースでは、広告は、一定のタイムリミットに達するまで又はページBがロードするまで放映及び再放映する連続ループである。後者のケースでは、ユーザは、活動が欠如する状態に直面することは決してない。

【0013】好ましくは、広告ウィンドウ(図8内のウィンドウ80を参照)内に表示される広告は、ページBがロードするのを待つ間ビューアを楽しませかつ情報を提供するアニメーションである。広告ウィンドウは同様に、広告対象となっている製品又はサービスについてのより多くの情報を受理するべく広告に関連するサイトまで「クリック・スルー」するようにユーザが広告ウィンドウ又はその一部分上でクリックすることができるように活動状態の制御機構をも内含してよい。特定の実施形態においては、広告サーバはユーザの人口統計データベースを維持し、データベース内に記憶されたそのユーザについての人口統計学的情報に基づいて送られた広告をターゲティングする。広告サーバは、ユーザの接続速度又はユーザが使用中の特定のブラウザについて広告

サーバが知っていることに応じて数多くの広告バージョンの中から1つの広告を選択することもできる。

【0014】上述のシステムによると、ページがロード中である間にクライアントユーザに何かすることを与えるということに加えて、いくつかの利点を得られる。多くのサーバオペレータが有利かつ好評なページを有するものの、ページの提供に見合うだけの充分な広告に欠けている。スポンサ付きサーバであることそして直接1つの広告サーバを、又間接的に望ましいページをポイントする広告リンクを内含することによって、スポンサ付きサーバは、広告サーバを通過するトラフィックに対する支払いを受けることができる。目標(ページB)ページはもう1つのスポンサ付きサーバ上にあってもよいし、又は独立サーバ上にあってもよい。スポンサ付きサーバであるということは、従来の「横断幕」広告主であるよりもサーバオペレータにとっては便利である。というのも、横断幕広告主は、ページ上の横断幕を適合させるためそのページを改訂する必要があるからである。本発明によると、広告は別のウィンドウ内で実行するため、表示はより単純である。好ましい実施形態においては、広告は、該当する場合、静止したビットマップと組合されて動くベクトル図形要素の形をしたアニメーションである。かかるアニメーションでは、図形要素の位置の変化は比較的少ないビットで指定でき、かくしてアニメーション用のデータを、広告の放映に必要な時間よりもはるかに短かい時間でダウンロードすることができる。

【0015】上述の説明は例示的なものであり、制限的意味はない。この開示を再度検討することにより当業者には、本発明の数多くの変形形態が明らかになることだろう。従って本発明の範囲は、上述の説明を参照して決定されるべきではなく、むしろ冒頭のクレームとその等価物の最大範囲を参考にして決定されるべきである。

付録A サンプルAd (Javaアプリケーション及びJavaアプレット)

1. Javaアプリケーション

【0016】

【表1】

```
import java.awt.*;
import java.applet.*;

public class SampleAd extends Frame implements Runnable {
    int frameNumber;
    Thread RunThread;

    //This is called from Ad.class
    SampleAd(String s)
    {
        setTitle(ldbquote Adletts Demordblquote );
        RunThread = new Thread(this);
        RunThread.start();
    }

    public void run() {
        long startTime;
        int delay;
        URL picloc;

        delay = 3000;

        startTime = System.currentTimeMillis();
        while (Thread.currentThread() == RunThread) {
            frameNumber++;

            if ( frameNumber == 6 ) {
                dispose();
            }

            toFront();
            repaint();

            try {
                startTime += delay;
                Thread.sleep(Math.max(0,
                    startTime-System.currentTimeMillis()));
            }
            catch (InterruptedException e) {
                break;
            }
        }
    }

    public void paint(Graphics g) {
        g.drawString("This is frame " + frameNumber, 5, 15);
    }
}
```

2. Java アプレット

【0017】

【表2】

```

import java.awt.*;
import java.applet.*;
import netscape.javascript.JSObject;

public class AppletAd extends Applet implements Runnable {
    int frameNumber;
    Thread RunThread;

    public void start() {
        RunThread = new Thread(this);
        RunThread.start();
    }

    public void run() {
        long startTime;
        int delay;
        URL picloc;

        delay = 3000;

        startTime = System.currentTimeMillis();
        while (Thread.currentThread() == RunThread) {
            frameNumber++;

            if (frameNumber == 6) {
                JSObject win = JSObject.getWindow(this);
                win.eval("self.close()");
            }

            repaint();

            try {
                startTime += delay;
                Thread.sleep(Math.max(0,
                    startTime - System.currentTimeMillis()));
            }
            catch (InterruptedException e) {
                break;
            }
        }
    }

    public void paint(Graphics g) {
        if (frameNumber == 1) {
            g.drawString("This is frame " + frameNumber, 5, 15);
        }
    }
}

```

著作権表示

本特許文書の開示の一部には、著作権保護の対象となっている資料が含まれている。著作権所有者は、特許文書又は特許の開示をそれが特許庁のファイル又は記録内に記されているものと正確に同じ形で何人かが電子写真複製することに対しいかなる異議を唱える者でもないが、それ以外の場合にはそれに対する全著作権を保有するものとする。

【図面の簡単な説明】

【図1】クライアントサーバがインターネット上に相互接続されているネットワーク化されたクライアント・サーバシステムのブロックダイアグラムである。

【図2】従来の拾い読みプロセスを例示するフローチャートである。

【図3】本発明に従った空き時間中に付加的情報が提供されている間の拾い読みプロセスを例示するフローチャ

ートである。

【図4】図4(a)は、クライアントブラウザ表示装置上に表示され得るHTMLページの例である。図4(b)は、クライアントブラウザ表示装置上に表示され得るHTMLページの例である。

【図5】本発明に従った、広告を開始するためのJavaアプレットを表わすHTMLページの図である。

【図6】目標ページまでブラウザを押すためのHTMLページの図である。

【図7】Javaアプリケーション起動ページを表わすHTMLページの図である。

【図8】1つのページがロードされている間の広告放映を伴うブラウザ表示の図である。

【符号の説明】

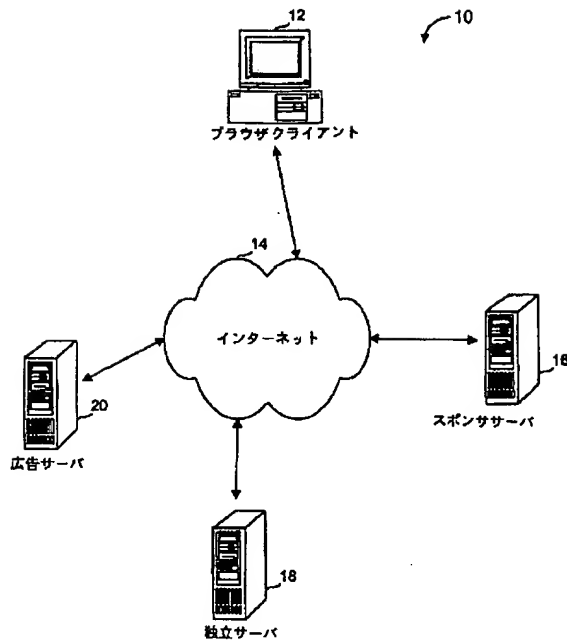
10-拾い読みシステム

12-ブラウザクライアント

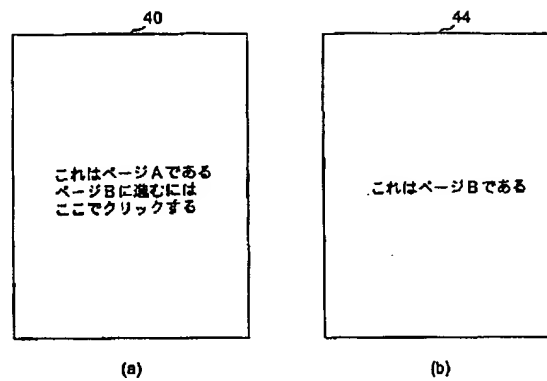
14-大域インターネット
16-スポンササーバ

18-独立サーバ
20-広告サーバ

【図1】



【図4】

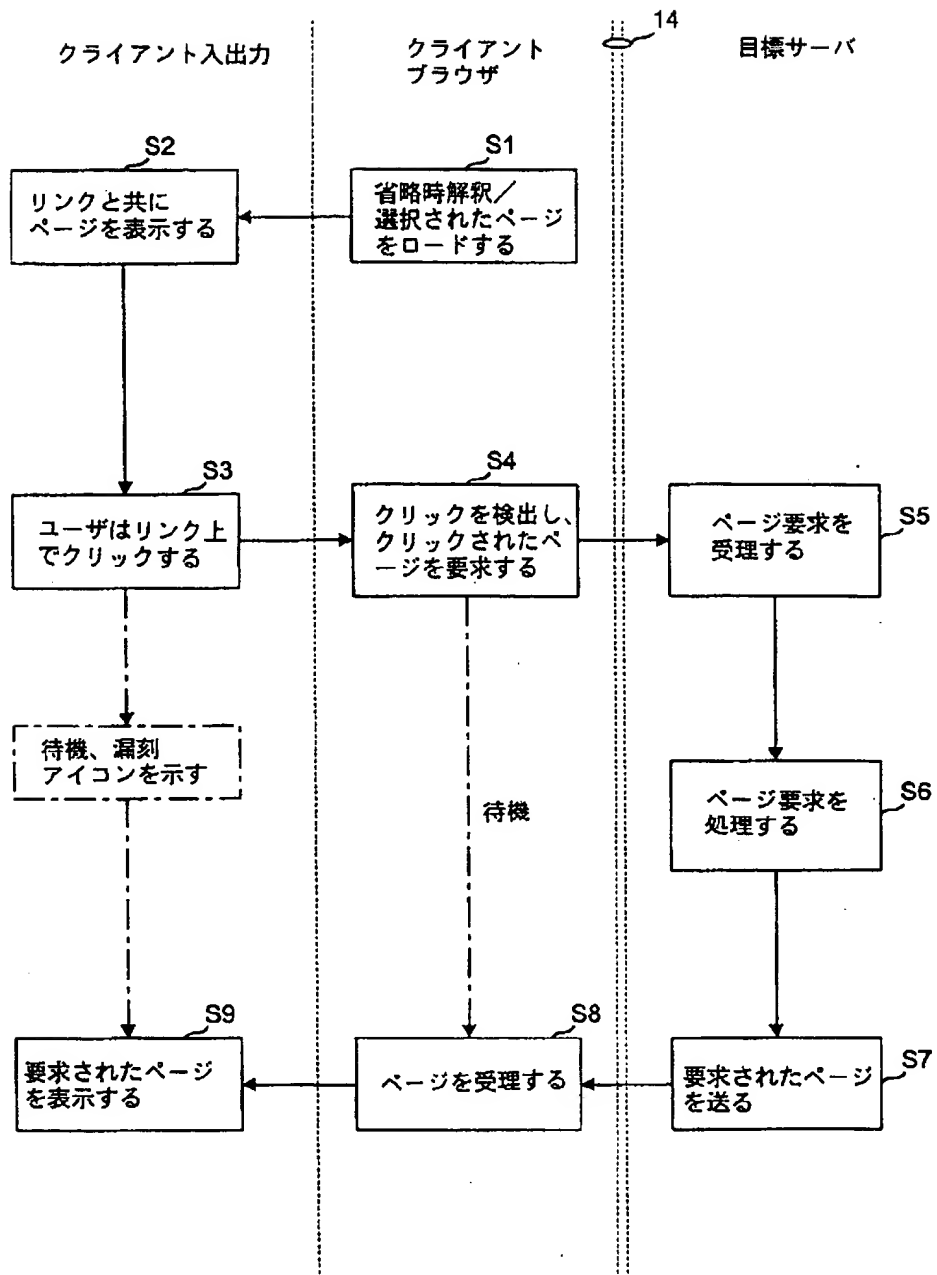


【図5】

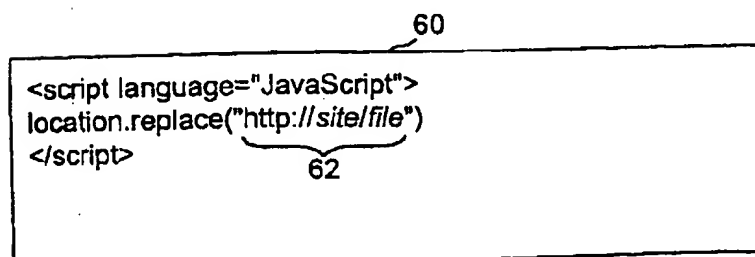
50

```
<script language="JavaScript">
  window.open("http://content.adletts.com/aag.dll?Ad=Today'sAd2&h=100&w=300",
    "", "scrollbars=no, toolbar=no, location=no, status=no, menubar=no, resizable=no,
    height=100, width=300
</script>
```

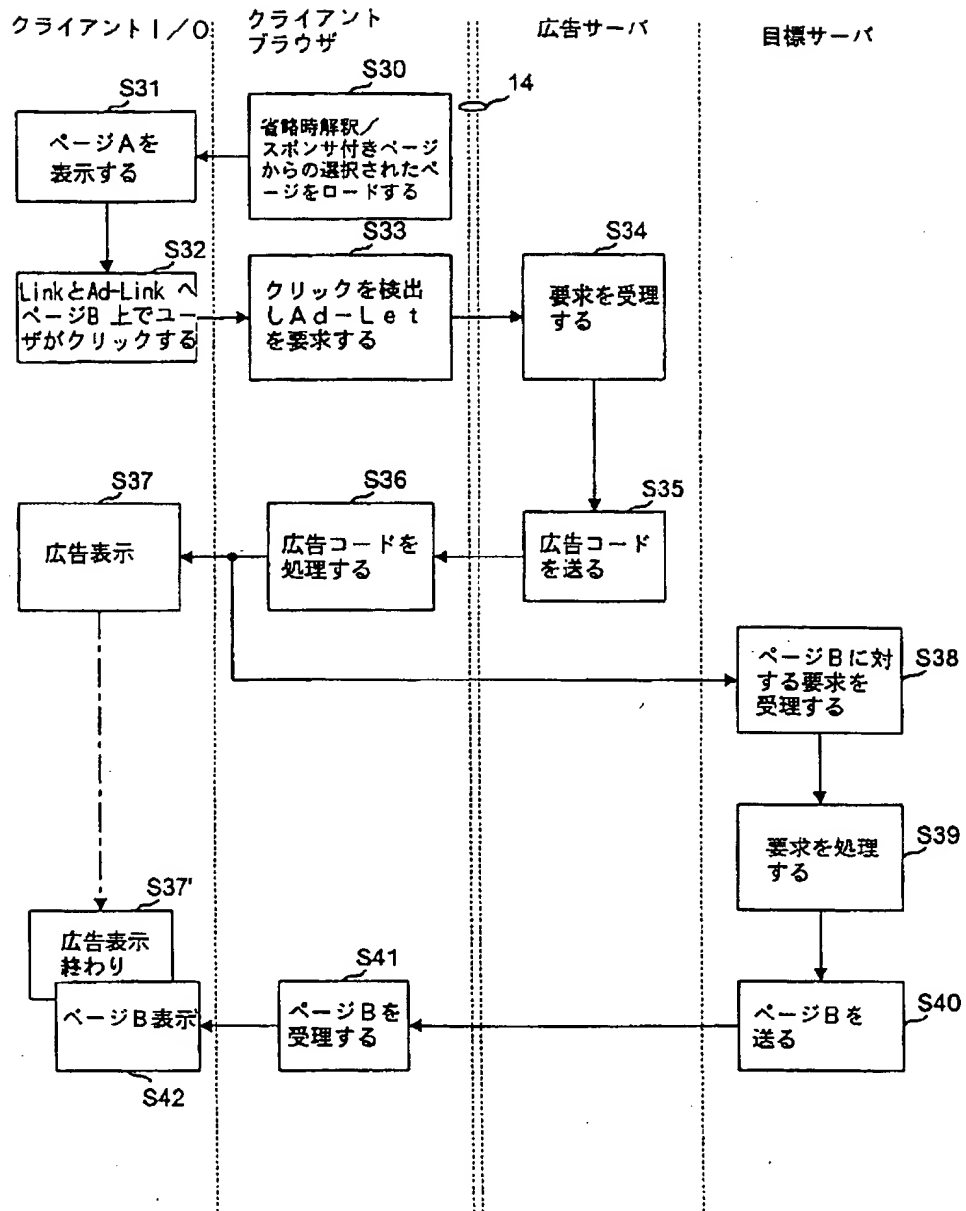
【図2】



【図6】



【図3】



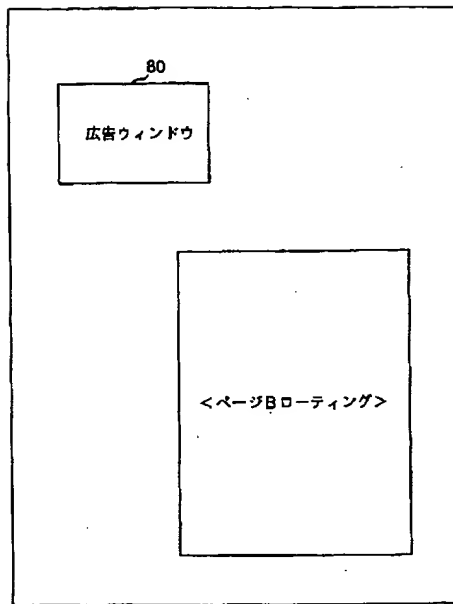
【図7】

70

```

<APPLET code="Ad.class" codebase="http://content.adl tts.com"
width=0 height=0><param name=ad value="TodaysAd1"></APPLET>
  
```

【図8】



フロントページの続き

(72)発明者 クリストファー リード
アメリカ合衆国 カリフォルニア州
94941 ミル ヴァリー エル シーデ
コート 1216